ABSTRACT

A scrambling architecture protects data streams in the operating system and hardware components of a computer by scrambling the otherwise raw data prior to the data being handled by the operating system. The architecture has a scrambler implemented at either the client or the server that adds noise to the content. More specifically, the scrambler produces periodic sets of tone patterns having varying amplitudes based on a first key. The scrambler also generates a random signal based on the first key and a second key. The tone patterns and random signal are added to the content to scramble the content. The scrambled content is then passed to the filter graph (or other processing system) where the content is processed while scrambled. Any attacker attempting to siphon off the bits during processing will steal only noisy data, which is worthless for redistribution or copying purposes. After processing, the scrambled data is passed to a driver for output. The driver implements a descrambler to unscramble the content by subtracting out the random noise signal. The descrambler detects the tone patterns in the content and recovers the first key from the varying amplitudes of the tone patterns. The descrambler also receives the second key via a separate channel (e.g., a cryptographically secured path) and generates the same random signal using the recovered first key and the second key. The descrambler subtracts the tone patterns and the random signal from the scrambled content to restore the content.